**Inflación global**

**Introducción:**

En el siguiente proyecto, voy a realizar un proceso de ETL a partir de un csv que contiene datos sobre la inflación de cada país desde el año 1980 hasta el 2024. El objetivo de dicho proyecto es mostrar la eficiencia de la integración de herramientas como Postgres-Docker-Airbyte-Snowflake.

**Pasos del proyecto:**

*Primer paso: realizar el proceso de ETL.*

En mi primer paso es crear una carpeta que contenga toda mi información sobre el proyecto que voy a realizar. Es por ello que a partir del uso del editor de código Visual Studio Code, creo un Jupyter Netbook bajo el nombre “*inflation\_etl.ipynb*”. Al principio de mi cuaderno de Jupyter lo primero es crear mi *Configuración de Entorno* de esta manera: Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Primero actualizo mi ***pip*** y luego instalo las bibliotecas que voy a utilizar en mi proyecto (el cual podrán encontrarlo en la carpeta de este).

Lo siguiente es cargar el conjunto de datos desde el archivo CSV utilizando pandas: Texto

Descripción generada automáticamente

Para verificar que se cargo correctamente y a su vez observar las características que presenta el mismo, voy a realizar una ***Exploración de Datos***.

*Subpaso uno: Exploración de Datos.*

Lo primero es visualizar como es mi conjunto de datos: Calendario

Descripción generada automáticamente

Con esto verificamos que la carga del CSV fue correcta y ahora procedo a analizar los distintos tipos de variables que contiene mi estructura de datos.

Lo primero es ver *que tipo de dato son las variables del conjunto de datos: Texto

Descripción generada automáticamente*

Como se visualiza la mayoria es de tipo ***float64*** y unicamente solo dos de tipo ***object***. Pero también quiero saber cuantos de tipo ***float64*** tengo, para ello: Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Entonces, tengo 45 de tipo ***float64***. Ahora me gustaría saber *cuantas filas y columnas tenemos en el conjunto de datos*: Texto

Descripción generada automáticamente

Por ende, en el dataframe tengo 196 filas y 47 columnas. Lo siguiente y fundamental es preguntarme si *existen valores nulos explícitos en el conjunto de datos*:

Texto

Descripción generada automáticamente

Forma

Descripción generada automáticamente con confianza media

Imagen que contiene Forma

Descripción generada automáticamenteComo se observa en todas las columnas de tipo ***float64*** se presentan valores nulos. A su vez me interesa saber cuantas variables nulas se presentan en cada una de esas columnas, para ello: Texto

Descripción generada automáticamente

Como se ve, después del año 2000 las variables nulas se hacen menos presentes, no obstante para visualizar de mejor manera la naturaleza de los valores nulos realizo lo siguiente: Texto

Descripción generada automáticamente

Si hacen correr este código, visualizaran el DataFrame mostrando en cada columna como los valores nulos están representados por la palabra ***NaN***. Es por ello que lo que se observa es que la naturaleza del mismo es producto de distintas causas, algunas son subsecuentes de cambios históricos como en el caso de Ucrania, el cual no se registra datos a partir de 1980 hasta años más adelante producto de la disolución de la Unión Soviética que ocasión que tanto Ucrania como otros países de la U.R.R.S. se formen como estado independiente. También otra naturaleza es debido a mejoras en la recopilación de datos o si los países quisieron compartir dicha información. Por consiguiente no voy aplicar ninguna acción sobre estos valores nulos.

*Subpaso dos: Transformación.*

En mi subpaso dos tengo distintos objetivos el cual son los siguientes: Texto

Descripción generada automáticamente

Para realizar mi primer objetivo:



Voy a escribir el siguiente código el cual me va a permitir crear un nuevo DataFrame que contenga todos los paises y los datos desde el 2000 hasta el 2023. También voy a usar la funcion ‘***melt()***’ para combinar las columnas de los años en una nueva columna llamada ‘*Year*’: Texto

Descripción generada automáticamente

A su vez me gustaría cambiar el nombre de las columnas “***country\_name***” por “***Country***” e “***indicator\_name***” por “***Indicator***”: Captura de pantalla de un celular

Descripción generada automáticamente

Ahora me toca completar mi segundo objetivo:



El cual es separar los países en distintos Datasets según sus continentes. Dejo abajo los siguientes códigos.

* ***Africa:***

CODIGO

“

country\_afr = [ "Algeria", "Angola", "Benin", "Botswana", "Burkina Faso", "Burundi", "Cabo Verde",

    "Cameroon", "Chad", "Comoros", "Democratic Republic of the Congo", "Djibouti", "Egypt",

    "Equatorial Guinea", "Eritrea", "Eswatini", "Ethiopia", "Gabon", "Gambia", "Ghana",

    "Guinea", "Guinea-Bissau", "Ivory Coast", "Kenya", "Lesotho", "Liberia", "Libya",

    "Madagascar", "Malawi", "Mali", "Mauritania", "Mauritius", "Morocco", "Mozambique",

    "Namibia", "Niger", "Nigeria", "Rwanda", "Sao Tome and Principe", "Senegal", "Seychelles",

    "Sierra Leone", "Somalia", "South Africa", "South Sudan", "Sudan", "Tanzania", "Togo",

    "Tunisia", "Uganda", "Zambia", "Zimbabwe", "North Sudan", "Western Sahara"]

infl\_dt\_1\_afr = infl\_dt\_1[infl\_dt\_1['Country'].isin(country\_afr)]

infl\_dt\_1\_afr

”

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

* ***América:***

CODIGO

“

country\_amer = ["Canada", "United States","Mexico", "Greenland", "Bermuda", "Saint Pierre and Miquelon","Belize", "Costa Rica", "El Salvador", "Guatemala", "Honduras", "Nicaragua", "Panama", "Antigua and Barbuda", "Bahamas", "Barbados", "Cuba", "Dominica", "Dominican Republic", "Grenada", "Haiti", "Jamaica", "Saint Kitts and Nevis", "Saint Lucia", "Saint Vincent and the Grenadines", "Trinidad and Tobago", "Puerto Rico", "Turks and Caicos Islands", "Cayman Islands", "British Virgin Islands", "United States Virgin Islands", "Anguilla", "Montserrat","Argentina", "Bolivia", "Brazil", "Chile", "Colombia", "Ecuador", "Guyana", "Paraguay", "Peru", "Suriname", "Uruguay", "Venezuela", "French Guiana"]

infl\_dt\_1\_amer = infl\_dt\_1[infl\_dt\_1['Country'].isin(country\_amer)]

infl\_dt\_1\_amer

”

Texto

Descripción generada automáticamente

* ***Asia:***

CODIGO

“

country\_asia = ["Afghanistan", "Armenia", "Azerbaijan", "Bahrain", "Bangladesh", "Bhutan","Brunei", "Cambodia", "China", "Cyprus", "Georgia", "India", "Indonesia","Iran", "Iraq", "Israel", "Japan", "Jordan", "Kazakhstan", "Kuwait","Kyrgyzstan", "Laos", "Lebanon", "Malaysia", "Maldives", "Mongolia","Myanmar", "Nepal", "North Korea", "Oman", "Pakistan", "Palestine","Philippines", "Qatar", "Saudi Arabia", "Singapore", "South Korea","Sri Lanka", "Syria", "Taiwan", "Tajikistan", "Thailand", "Turkmenistan","United Arab Emirates", "Uzbekistan", "Vietnam", "Yemen"]

infl\_dt\_1\_as = infl\_dt\_1[infl\_dt\_1['Country'].isin(country\_asia)]

infl\_dt\_1\_as

”

Texto

Descripción generada automáticamente

* ***Europa:***

CODIGO

“

country\_eu = ["Albania", "Andorra", "Armenia", "Austria", "Azerbaijan", "Belarus", "Belgium","Bosnia and Herzegovina", "Bulgaria", "Croatia", "Cyprus", "Czech Republic","Denmark", "Estonia", "Finland", "France", "Georgia", "Germany", "Greece","Hungary", "Iceland", "Ireland", "Italy", "Kazakhstan", "Kosovo", "Latvia", "Liechtenstein", "Lithuania", "Luxembourg", "Malta", "Moldova", "Monaco","Montenegro", "Netherlands", "North Macedonia", "Norway", "Poland", "Portugal", "Romania", "Russia", "San Marino", "Serbia", "Slovakia","Slovenia", "Spain", "Sweden", "Switzerland", "Turkey", "Ukraine", "United Kingdom", "Vatican City", "Faroe Islands", "Gibraltar"]

infl\_dt\_1\_eu = infl\_dt\_1[infl\_dt\_1['Country'].isin(country\_eu)]

infl\_dt\_1\_eu

”

Texto

Descripción generada automáticamente

* ***Oceanía***

CODIGO

“

country\_ocean = ["Australia", "Fiji", "Marshall Islands", "Solomon Islands", "Kiribati", "Micronesia", "Nauru", "New Zealand", "Palau", "Papua New Guinea","Samoa", "Tonga", "Tuvalu", "Vanuatu"]

infl\_dt\_1\_oc = infl\_dt\_1[infl\_dt\_1['Country'].isin(country\_ocean)]

infl\_dt\_1\_oc

”

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Ahora en mi tercer objetivo

**

Para ello escribo el siguiente código el cual me permita crear la nueva columna y a su vez, guarde esa información de cada continente en nuevos datasets el cual posteriormente se guardaron en un CSV en la carpeta con el mismo nombre.

CODIGO

“

# AFRICA

infl\_dt\_1\_afr['Continent'] = 'AFR'

df\_infl\_dt\_afr = pd.DataFrame(infl\_dt\_1\_afr)

df\_infl\_dt\_afr.to\_csv('./csv/infl\_dt\_afr.csv')

# AMERICA

infl\_dt\_1\_amer['Continent'] = 'AMER'

df\_infl\_dt\_amer = pd.DataFrame(infl\_dt\_1\_amer)

df\_infl\_dt\_amer.to\_csv('./csv/infl\_dt\_amer.csv')

# ASIA

infl\_dt\_1\_as['Continent'] = 'AS'

df\_infl\_dt\_as = pd.DataFrame(infl\_dt\_1\_as)

df\_infl\_dt\_as.to\_csv('./csv/infl\_dt\_as.csv')

# EUROPA

infl\_dt\_1\_eu['Continent'] = 'EU'

df\_infl\_dt\_eu = pd.DataFrame(infl\_dt\_1\_eu)

df\_infl\_dt\_eu.to\_csv('./csv/infl\_dt\_eu.csv')

# OCEANIA

infl\_dt\_1\_oc['Continent'] = 'OC'

df\_infl\_dt\_oc = pd.DataFrame(infl\_dt\_1\_oc)

df\_infl\_dt\_oc.to\_csv('./csv/infl\_dt\_oc.csv')

”

Ahora en nuestro último objetivo



Voy a unir esos datasets en uno nuevo, para ello creo una nueva lista llamada “*data\_frames*” que contendrá todos mis DataFrames creados anteriormente, y luego los combino en un mismo DataFrame.

CODIGO

“

# Creo una lista llamada data\_frames que contendra todos mis DataFrames creados

data\_frames = [df\_infl\_dt\_afr, df\_infl\_dt\_amer, df\_infl\_dt\_as, df\_infl\_dt\_eu, df\_infl\_dt\_oc]

# Combino todos los DataFrames en uno solo

inflation\_data = pd.concat(data\_frames, ignore\_index= True)

inflation\_data

”

*Tercer subpaso: Carga.*

En nuestro ultimo paso de mi proceso de ETL tengo que realizar la carga del CSV que contendrá la informacion de todas las transformaciones que realice en mi CSV original. Para ello, guardo el DataFrame que cree anteriormente en un nuevo CSV con destino en la carpeta de dicho nombre.

CODIGO

“

df = pd.DataFrame(inflation\_data)

df.to\_csv('./csv/inflation\_data.csv')

”

*Segundo paso: desplegar postgres a través de Docker.*

En este paso voy a crear un archivo “.yml” bajo el nombre “*postgres\_docker.yml*”. Texto

Descripción generada automáticamente

Este archivo contiene la configuración para desplegar un contenedor de PostgreSQL utilizando Docker Compose. El código en el archivo YAML define cómo se creara y configurara el contenedor de PostgreSQL, incluyendo detalles como la imagen de Docker a utilizar “***postgres:latest***”, el nombre del contenedor, la configuración de las variables de entorno (como la contraseña de la base de datos), los puertos que se expondrán, y los volúmenes que se montarán. A continuación presento el código que esta dentro del archivo mencionado

CODIGO

“

version: '3.1'

services:

  postgres:

    image: postgres:latest

    container\_name: inflation\_postgres

    restart: always

    environment:

      POSTGRES\_PASSWORD: admin

    ports:

      - "5432:5432"

    volumes:

      - pgdata:/var/lib/postgresql/data

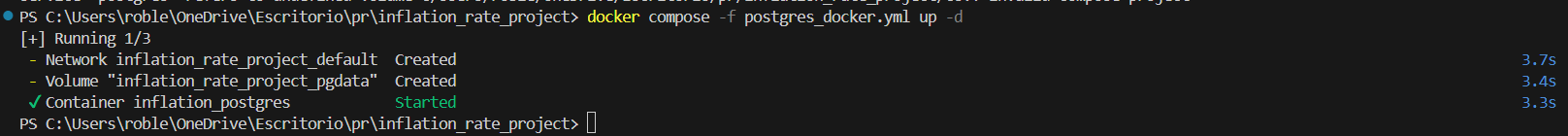
      - /c/Users/roble/OneDrive/Escritorio/pr/inflation\_rate\_project/csv:/inflation\_csv

volumes:

  pgdata:

”

Ahora en la terminal de mi editor de código, escribo el siguiente comando.



Esto me permitirá iniciar los servicios definidos en el archivo anteriormente mencionado utilizando Docker Compose, y ejecutarlos en segundo plano. Por ende, esto iniciará el contenedor de PostgreSQL definido en el archivo YAML.

Ahora nos metemos en Docker y buscamos el contenedor creado. Dentro del mismo utilizo su terminal para conectarme a postgres bajo el comando:

CODIGO

“psql -U postgres”

Texto

Descripción generada automáticamente

Una vez dentro, creo un database:

CODIGO

“CREATE DATABASE INFLATION\_GLOBAL;”



Con nuestra Database creada, solo nos toca crear nuestras dos tablas que contendrán por un lado información de la inflación a nivel global y otra a nivel continental. Entonces, ingresamos a nuestra database:

CODIGO

“\c inflation\_global;”



Una vez dentro, creamos la tabla para la inflación global:

CODIGO

  CREATE TABLE infl\_global (

    Id SERIAL PRIMARY KEY,

Country VARCHAR(255),

    Indicator VARCHAR(50),

    Year INT,

    Inflation FLOAT,

Continent VARCHAR(255)

);

Texto

Descripción generada automáticamente

Ahora que cree la tabla, coloco la información respectiva dentro de la misma. Dicha información esta contenida en el CSV “inflation\_data.csv”

CODIGO

“\COPY infl\_global FROM ‘./inflation\_csv/inflation\_data.csv’ DELIMITER ‘,’ CSV HEADER;”



DATA PARA LOS DATOS CONTINENTALES

Una vez que cree la tabla para la inflación a nivel global y cargue su respectiva información, voy a desconectarme del database para volver al principio de la terminal de postgres con el fin, de crear otro database bajo el nombre “*inflation\_continent*”. Entonces el código para crear el nuevo database y a su vez conectarme:

CODIGO

“

CREATE DATABASE INFLATION\_CONTINENT;

”

“

\c inflation\_continent;

”

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Además vez voy a crear varias tablas con el nombre de los continentes y cargare su información respectiva.

Entonces voy a escribir los siguientes códigos para la creación de las tablas:

  CREATE TABLE inflation\_africa (

    Id SERIAL PRIMARY KEY,

Country VARCHAR(255),

    Indicator VARCHAR(50),

    Year INT,

    Inflation FLOAT,

Continent VARCHAR(255)

);

  CREATE TABLE inflation\_america (

    Id SERIAL PRIMARY KEY,

Country VARCHAR(255),

    Indicator VARCHAR(50),

    Year INT,

    Inflation FLOAT,

Continent VARCHAR(255)

);

  CREATE TABLE inflation\_asia (

    Id SERIAL PRIMARY KEY,

Country VARCHAR(255),

    Indicator VARCHAR(50),

    Year INT,

    Inflation FLOAT,

Continent VARCHAR(255)

);

  CREATE TABLE inflation\_europa (

    Id SERIAL PRIMARY KEY,

Country VARCHAR(255),

    Indicator VARCHAR(50),

    Year INT,

    Inflation FLOAT,

Continent VARCHAR(255)

);

  CREATE TABLE inflation\_oceania (

    Id SERIAL PRIMARY KEY,

Country VARCHAR(255),

    Indicator VARCHAR(50),

    Year INT,

    Inflation FLOAT,

Continent VARCHAR(255)

);

Una vez creada las tablas escribo los siguientes códigos para cargar la información respectiva a cada una:

CODIGO

“\COPY inflation\_africa FROM ‘./inflation\_csv/infl\_dt\_afr.csv’ DELIMITER ‘,’ CSV HEADER;”

“\COPY inflation\_america FROM ‘./inflation\_csv/infl\_dt\_amer.csv’ DELIMITER ‘,’ CSV HEADER;”

“\COPY inflation\_asia FROM ‘./inflation\_csv/infl\_dt\_as.csv’ DELIMITER ‘,’ CSV HEADER;”

“\COPY inflation\_europa FROM ‘./inflation\_csv/infl\_dt\_eu.csv’ DELIMITER ‘,’ CSV HEADER;”

“\COPY inflation\_oceania FROM ‘./inflation\_csv/infl\_dt\_oc.csv’ DELIMITER ‘,’ CSV HEADER;”

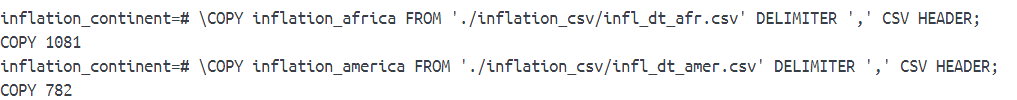
Así queda finalizado este segundo paso donde nos conectamos a postgres para cargar los database con sus tablas e información respectiva.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente con confianza baja

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

*Tercer paso: preparar Snowflake para nuestra conexión con Airbyte.*

Ahora voy a utilizar la herramienta Snowflake para que los datos que más adelante se transfieran a través de Airbyte queden en las distintas tablas de los databases que voy a crear en esta herramienta. Para ello lo primero es ingresar a su respectiva página (<https://www.snowflake.com/es/>)

Y luego nos dirigimos a:

En este paso mediante el uso de el “*Schema*” voy a configurar el código que proporciona Airbyte en su pagina para que podamos configurar nuestro ambiente para la posterior conexión que voy a realizar en el siguiente paso.

Ingresamos a la siguiente pagina de Airbyte:

<https://airbyte.com/how-to-sync/postgresql-to-snowflake-data-cloud>

Y nos dirigimos a esta sección: Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

Snowflake nos ofrece una cuenta gratis por 30 dias ideal para hacer proyectos.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Ingresamos nuestros datos y hacemos click en continuar. Inmediatamente Snowflake nos enviará un email con varios datos importantes, pero el que más importa es el siguiente:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Esto hay que tenerlo en cuenta para su uso posteriormente.

Ahora ingresamos a nuestra cuenta de Snowflake y vamos a observar el siguiente apartado:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Hacemos click en: Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

Y copiaremos el siguiente código:

“

-- set variables (these need to be uppercase)  
set airbyte\_role = 'AIRBYTE\_ROLE';  
set airbyte\_username = 'AIRBYTE\_USER';  
set airbyte\_warehouse = 'AIRBYTE\_WAREHOUSE';  
set airbyte\_database = 'AIRBYTE\_DATABASE';  
set airbyte\_schema = 'AIRBYTE\_SCHEMA';  
‍

-- set user password  
set airbyte\_password ='YOUR\_AIRBYTE\_PASSWORD';  
  
begin;  
  
-- create Airbyte role  
use role securityadmin;  
create role if not exists identifier($airbyte\_role);  
grant role identifier($airbyte\_role) to role SYSADMIN;  
  
-- create Airbyte user  
create user if not exists identifier($airbyte\_username)  
password = $airbyte\_password  
default\_role = $airbyte\_role  
default\_warehouse = $airbyte\_warehouse;  
  
grant role identifier($airbyte\_role) to user identifier($airbyte\_username);  
  
-- change role to sysadmin for warehouse/database steps  
use role sysadmin;  
  
-- create Airbyte warehouse  
create warehouse if not exists identifier($airbyte\_warehouse)  
warehouse\_size = xsmall  
warehouse\_type = standard  
auto\_suspend = 60  
auto\_resume = true  
initially\_suspended = true;  
  
-- create Airbyte database  
create database if not exists identifier($airbyte\_database);  
  
-- grant Airbyte warehouse access  
grant USAGE  
on warehouse identifier($airbyte\_warehouse)  
to role identifier($airbyte\_role);  
  
-- grant Airbyte database access  
grant OWNERSHIP  
on database identifier($airbyte\_database)  
to role identifier($airbyte\_role);  
  
commit;  
  
begin;  
  
USE DATABASE identifier($airbyte\_database);  
  
-- create schema for Airbyte data  
CREATE SCHEMA IF NOT EXISTS identifier($airbyte\_schema);  
  
commit;  
  
begin;  
  
-- grant Airbyte schema access  
grant OWNERSHIP  
on schema identifier($airbyte\_schema)  
to role identifier($airbyte\_role);  
  
commit;

”

Esta línea de código de Airbyte nos proporciona un script para ejecutar esta autmatización. Por mi parte, modifique las primeras líneas que vera en la imagen de abajo, creando mi database, usuario, y los schema que más adelante me servirán para producir la sincronización.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Una vez que realice los cambios pertinentes ejecuto el script.

*Cuarto paso: Conexión con Airbyte.*

Para conectarme a Airbyte lo único que tengo que realizar es descargarme a través de github su repositorio. Para ello, ingresamos en la siguiente página:

<https://github.com/airbytehq/airbyte>

Y procedemos a descargarlo manualmente o a través de github. Si es a través de github, primero ingresamos a esta aplicación y dentro de ahí nos dirigimos a nuestra carpeta del proyecto. Una vez dentro escribimos lo siguiente:

CODIGO

“

git clone --depth=1 https://github.com/airbytehq/airbyte.git

”

Una vez que se nos descargue el repositorio, nos dirigimos a la carpeta

CODIGO

“

cd airbyte

”

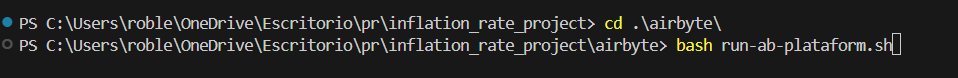
Y ahora escribimos lo siguiente:

CODIGO

“

bash run-ab-platform.sh

”



Esto generara que se descarguen varios archivos docker por si mismo como también contenedores para que nos pueda ejecutar correctamente la plataforma Airbyte. Como ven, ya tengo los contenedores Airbyte corriendo:

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente con confianza media

Ahora me dirijo a Google y escribo <http://localhost:8000>. Una vez dentro, hago click en el apartado “*New connection*” Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

En este apartado me permitirá configurar mi nueva conexión entre mis fuentes de datos y el destino de estos datos. Generalmente, me pedirá que elija mi fuente de datos que deseo conectar, y una vez seleccionada, voy a propiciar los detalles de autenticación y configuración necesario para conectarme a esa fuente de datos. Una vez que la conexión se ha establecido correctamente, voy a configurar la sincronización de datos, programar actualizaciones periódicas y mapear los datos a la estructura adecuada en mi destino de datos. Por ende, empecemos con seleccionar primero mi fuente de datos el cual es postgres.Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

Una vez elegida nos mostrara una serie de configuraciones el cual debemos completar usando las siguientes características:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Una vez escogida nuestra fuente de datos y la configuración de la misma, voy a proceder a seleccionar el destino de mis datos y su configuración pertinente. Entonces, escojo SnowflakeInterfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

Y su configuración será la siguiente. Recuerden que anteriormente les había mencionado que en el email que Snowflake les envió una vez que nos hemos registrado, había una parte que decía “*Dedicated Login URL*”. Lo seguido a eso lo copian y lo pegan en la parte que dice “*Host*” en el formulario de configuración. Ahora tanto *Warehouse*, como *Role, Database, Default\_Schema y Username*, se encuentra en esta sección:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Solamente tenemos que copiar lo mismo que nos dice en cada sección. Una vez hecho eso, en la parte que menciona “*Autorization Method*” lo cambiamos por el método “*Username and Password*” y ahí colocamos la contraseña que configuramos en el script de arriba, en la sección “*set airbyte\_password*”.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico, Teams

Descripción generada automáticamente

Una vez configurado damos a continuar. Y por último nos pedira la configuración de la conexión. Para ello, seleccionamos “*Schedule type*” y lo colocamos en “*Manual*”. En el apartadod de “*Activate the streams you want to sync*” observamos nuestro schema que creamos en la terminal de postgres, significa que vas por buen camino. Ahora solo toca en hacer click en “*Set up connection*”

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteImagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Una vez configurado la configuración nos pedirá sincronizar ahora los datos con lo cual hacemos click en ello.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteUna vez que se nos sincronicen los datos, nos dirigimos a Snowflake y en el apartado “*Database*” y seleccionamos nuestro database “*Inflation*”. Dentro del mismo encontraremos nuestro schema “*INFLATION\_GLOBAL*” y veremos que está cargado la tabla “*INFL\_GLOBAL*” con toda nuestra información subida.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Lo último que tenemos que realizar ahora es sincronizar nuestros datos de la tabla que contiene la inflación a nivel continental. Entonces desde la pagina principal de Airbyte nuevamente selecciono “*New connection*” y copiamos lo siguiente:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ahora copio lo siguiente para configurar nuestra conexión con Snowflake:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación, Correo electrónico, Teams

Descripción generada automáticamente

Y por último, configuramos nuestra conexión y seguido, sincronizamos tanto las tablas como su información hacia nuestro schema “*INFLATION\_GLOBAL*” en Snowflake.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación

Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

Nos dirigimos a Snowlfake y verificamos que nuestras tablas como nuestra informacion se hayan sincronizado.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Como podemos ver hemos terminado el proyecto cumpliendo con el objetivo propuesto al principio.